CLAIMS

What is claimed is:

1. A method, comprising:

analyzing a first code; and

generating a second code based on the first code, the second code including a microarchitecture implementation-specific alternative representation of at least some portions of the first code.

2. The method of claim 1, wherein the first code comprises source code.

3. The method of claim 1, wherein the first code comprises code compiled from source code for a given instruction set architecture (ISA).

4. The method of claim 3, wherein the second code further comprises the first code.

5. The method of claim 4, further comprising generating boundary markers to mark a beginning and an end for the alternative representation, the boundary markers being in a format of the ISA.

6. The method of claim 5, further comprising generating a trigger instruction which when executed by a machine executing the second code causes the machine to execute the alternative representation instead of the first code, the trigger being in the format of the ISA.

7. The method of claim 1, wherein the alternative representation comprises microcode.

8. A machine-readable medium having stored thereon a code sequence, comprising:

compiled code for a given ISA; and

discrete regions of microarchitecture implementation-specific code bounded by ISA format markers.

9. The machine-readable medium of claim 8, wherein the microarchitecture implementation-specific code comprises an alternative representation of the compiled code.

10. The machine-readable medium of claim 9, wherein the code sequence further comprises a trigger instruction in the format of the ISA which when executed by hardware causes the microarchitecture implementation-specific code to be executed instead of the compiled code for the ISA.

11. A microprocessor, comprising:

a fetch unit; and

a decode unit, the microprocessor having first and second modes of operation, wherein in the first mode, the decode unit decodes ISA format instructions supplied by the fetch unit, and in the second mode, the decode unit processes microarchitecture implementation-specific format instructions supplied by the fetch unit.

12. The microprocessor of claim 11, wherein transitions between the first and second modes occur upon detection by the decode unit of an ISA format boundary marker supplied by the fetch unit.

13. The microprocessor of claim 11, wherein processing the microarchitecture implementation-specific format instruction comprises decoding the instruction.

14. A method, comprising:

analyzing a first code comprising instructions for a first ISA; and

generating a second code based on the first code, the second code including at least some instructions for a second ISA corresponding to instructions in the first code.

15. The method of claim 14, wherein the second code comprises microinstructions.

16. The method of claim 14, wherein the first code is for execution by a first processing unit; and the second code is for execution by a second processing unit which supports a different ISA from the first processing unit.

17. A machine-readable medium having stored thereon a sequence of instructions which when executed by a processor, cause the processor to perform a method comprising:

analyzing a first code; and

generating a second code based on the first code, the second code including a microarchitecture implementation-specific alternative representation of at least some portions of the first code.

18. The machine-readable medium of claim 17, wherein the first code comprises source code.

19. The machine-readable medium of claim 17, wherein the first code comprises code compiled from source code for a given ISA.

20. A system, comprising:

a memory; and

a microprocessor coupled to the memory, the microprocessor including a fetch unit and a decode unit, wherein the microprocessor has first and second modes of operation, wherein in the first mode, the decode unit decodes ISA format instructions supplied by the fetch unit, and in the second mode, the decode unit processes microarchitecture implementation-specific format instructions supplied by the fetch unit.

21. The system of claim 20, wherein transitions between the first and second modes occur upon detection by the decode unit of an ISA format boundary marker supplied by the fetch unit.

22. A computer-readable medium having stored thereon a sequence of instructions which when executed by the processor, cause the processor to perform a method comprising:

analyzing a first code comprising instructions for a first ISA; and

generating a second code based on the first code, the second code including at least some instructions for a second ISA format corresponding to instructions in the first code.

23. The computer-readable medium of claim 22, wherein the second code comprises microinstructions.

24. The system of claim 22, wherein the first code is for execution by a first processing unit; and the second code is for execution by a second processing unit which support a different ISA from the first processing unit.

25. A system, comprising:

a processor; and

a memory coupled to the processor, the memory storing instructions which are executed by the processor, cause the processor to perform a method comprising;

analyzing a first code; and

generating a second code based on the first code, the second code including a microarchitecture implementation-specific alternative representation of at least some portions of the first code.

26. The system of claim 25, wherein the first code comprises source code.

27. The system of claim 26, wherein the first code comprises code compiled from source code from a given ISA.

28. A system, comprising:

a processor; and

a memory coupled to the processor, the memory storing instructions which are executed by the processor cause the processor to perform a method comprising;

analyzing the first code comprising instructions for a first ISA;

generating a second code based on the first code, the second code including at least some instructions for a second ISA format corresponding to instructions in the first code.

29. The system of claim 28, wherein the second code comprises microinstructions.